REMARKS

We would like to thank the Examiner for granting the very productive interview of February 17, 2005 for this application. Claims 10-21 are pending in this application. Claim 21 is a new claim.

The Examiner has objected to dependent claims 13 and 15-17 as indefinite because certain terms lack appropriate antecedent basis. Applicants have corrected claims 13 and 15-17 to address the antecedent basis issues. The examiner has also rejected claims 10-20 under 35 U.S.C. § 101 because the claimed invention is directed toward non-statutory subject matter. Applicants have amended claims 10-20 to address the §101 rejections.

The Examiner has also rejected claims 10-20 under 35 U.S.C. §102(b) as being anticipated by Luis De Rose et al, *A MATLAB to FORTRAN 90 Translator and Its Effectiveness*, ACM ("De Rose"). Applicants respectfully traverse this rejection and request reconsideration and allowance of claims 10-20 for the following reasons.

Applicants respectfully submit that Rose neither discloses nor suggests the pending claims as amended herein. According to M.P.E.P §706.02, "for anticipation under 35 U.S.C. 102, the reference must teach *every aspect* of the claimed invention...." (emphasis added). However, Rose fails to disclose or suggest various elements of the claimed invention.

De Rose discloses a method for determining the size of MATLAB variables using a combination of static and dynamic strategies. De Rose, p. 309. The De Rose static mechanism utilizes explicit declarations of variable size, whereas the dynamic strategies are employed at run-time to infer variable size when none is explicitly declared. *Id.* Thus, unlike the present invention which determines all variable sizes prior to run-time, the De Rose system must rely on

a type of dynamic determination in which the shape of the variables is not known until the program is actually executed. *See*, e.g. De Rose, p. 309 ("dynamic strategies that are applied at execution time when a lack of statically available information prevents automatic generation of a particular variable's declaration.")

De Rose's dynamic strategy for determining the shape of variables at run-time uses a common prior art technique that inserts source code which includes shadow variables into the resulting FORTRAN program. De Rose, p. 312-13. This inserted source code and its associated shadow variables keep track of the size and shape of variables as the FORTRAN program executes, i.e. during run-time. *Id.* The shadow variable method described in De Rose has several disadvantages: (1) the De Rose method introduces significant overhead by inserting extra source code into the FORTRAN program which results in greater demands on memory and longer execution times; (2) the De Rose method only works where additional memory resources can be dynamically allocated during run time; (3) the De Rose method does not determine the size of all variables prior to run-time; and (4) using the De Rose method, each program statement is treated in isolation, and as a result, the determined size of a variable is not propagated across all program statements using that variable.

The present invention addresses these problems, and specifically distinguishes the De Rose method in the specification. Appl. pp., 3-4, 7. The present invention uses shape-tuple notation and shape-tuple operators to determine the size and shape of variables prior to run-time. Appl. p. 7. These distinctions are recited in the claims as amended herein. For example, claim 10 recites "***determining an input shape-tuple*** for each operand of a program expression...wherein the ***size of at least one operand is unknown.***" Because the De Rose method neither uses shape-tuples, nor determines the shape or size of each operand, this element alone distinguishes the claims from De Rose. Additionally, claim 10 recites, "***determining prior***

to run-time a resulting shape-tuple of the program expression." Because De Rose depends upon source code inserted into the FORTRAN program to determine size of variable while the program is running, De Rose cannot determine the shape or size of those variables prior to run-time. This element of claim 10 clearly recites that the invention determines the resulting shape-tuples ***prior to run-time.*** Therefore, this element, among others, also distinguishes the claim from the De Rose method.

Claims 11-20 are dependent upon Claim 10, and therefore inherit the limitations of Claim 10. Hence claims 11-20 are distinguishable over De Rose for at least the reasons mentioned above.

Conclusion

In sum, Applicants respectfully submit that claims 10-21 as presented herein, are patentably distinguishable over the cited references (including references cited, but not applied). Therefore, Applicants request reconsideration and allowance of these claims.

Applicants respectfully invite Examiner to contact Applicants' representative at the number provided below if Examiner believes it will help expedite furtherance of this application.

RESPECTFULLY SUBMITTED,
Pramod G. Joisha, Prithviraj Banerjee, Nagaraj Shenoy

Date: 03/07/05

By: Deepti Pa[illegible]

Deepti Panchawagh-Jain, Esq.
Registration No. 43,846
3039 Calle De Las Estrella
San Jose, CA 95148
(408) 274-3043 (Phone)
(408) 516-5825 (Fax)